

"Express Mail" mailing label number:

EV 335895489 US

## **VOICEXML AND RULE ENGINE BASED SWITCHBOARD FOR INTERACTIVE VOICE RESPONSE (IVR) SERVICES**

Hisao M. Chang

### **Field of the Disclosure**

[0001] This disclosure relates generally to telecommunication systems in both public or private circuit-switched telephone networks (PSTN) and Internet-based Voice-over-IP (VoIP) networks

### **BACKGROUND**

[0002] Certain early work in call routing technologies are based on previous generations of interactive voice response (IVR) platforms where sharing a centralized business logic and decision source among hundreds of isolated IVR systems within an enterprise is not technically feasible. When calls come to one IVR node, the business logic that decides where to route the call to another IVR node or live agent is hardwired into the service logic being executed on that node. It is difficult to maintain the consistency or update such business logic for any enterprise with a large number of IVR platforms installed in multiple locations.

[0003] More recent systems use automatic call distribution (ACD) using a proprietary application programming interface (API) in conjunction with various computer telephony interface (CTI) technologies. The routing logic in the ACD is hardcoded in software programs running on the ACD. Similarly, when one IVR decides to send the call to another IVR or to a live agent via CTI, it cannot access up-to-the-minute business logic in order to determine the best routing strategy. Instead, individual IVRs rely on their own routing table which reside on individual servers. When there are multiple ACD nodes and dozens of IVR platforms across an enterprise call center environment, it is very

difficult to maintain a common image across all of these routing tables to reflect a centralized business logic on a real-time basis.

[0004] Accordingly, there is a need for an improved system and method of handling calls in an enterprise call center environment.

### **DETAILED DESCRIPTION**

[0005] The present application discloses a call routing system and a method of communicating with a call originator. The call routing system includes a voice converted data module having an input to receive an incoming call, an interactive voice response dialog module responsive to the voice converted data module; and a call routing module responsive to the voice converted data module. The call routing module is to route the incoming call to a destination. The method of communicating with an originator of a call includes receiving a call at an automated call handling system; performing an evaluation of the call based on a set of business rules; routing the call to an interactive voice response unit based on the evaluation, and in response to the call, automatically scheduling and sending an email to the originator of the call. The email includes a targeted communication message relating to the subject matter of the call.

[0006] The disclosed system provides a VoiceXML based software-driven switchboard powered by multiple rule engines that routes calls from one source to another. Referring to FIG. 1, a system for handling calls is shown. The system includes a voice converted data module (e.g. VoiceXML root engine 100), an IVR dialog engine 120, and a routing engine 130. The VoiceXML root engine 100, the IVR dialog engine 120, and the routing engine 130 are all coupled to an application server 170. The application server 170 is coupled to a database 116 to store business rules and logic. The VoiceXML root engine 100 is coupled to a second database 114 which includes DNIS rule tables. The VoiceXML root engine 100 is coupled to the application server via connection 112 and is coupled to the DNIS rule table 114 via a connection 110. The VoiceXML root engine 100 is coupled to the IVR dialog engine 120 via connection 113. The VoiceXML root engine 100 has an input responsive to various incoming IVR applications 106 (IVR1, IVR2 to IVRn) and internet-based telephony systems such as the session internet protocol

(SIP) phone 102. The IVR dialog engine 120 is coupled to a computer telephony interface (CTI) 140 via connection 121. The computer telephony interface (CTI) 140 is coupled to a customer relationship management (CRM) database 142 via connection 141. Connection 141 may carry a telephone number request and may retrieve customer information related to that telephone number from the CRM 142.

[0007] The CTI 140 is coupled to an agent terminal 147 via connection 146 and is coupled to a personalized call hold queue 102. The personalized queue 102, is routed to agent queues 104 having access to an audio content library 154. The agent queue 104 may be routed to particular types of automated call handling systems, such as billing system 156, repair system 157, and collection system 158. These subject matter based call handling systems may route calls to an agent terminal to provide customer assistance. Alternatively, the subject matter call handling systems may be connected to interactive voice response units to provide automated and computer generated responses to customer inquiries.

[0008] The routing engine 130 is coupled to an internet connection 150 via web agent 134 and connection 132. The web agent 134 may provide a telephone number 141 to customer relationship management database 142 for the retrieval of additional information associated with the telephone number for a given customer. An example of such information includes a customer history, prior transactions, address, name and call profile preferences. The web agent 134, via the web interface 150, may provide electronic communication such as email notifications 160. The routing engine module 130 is coupled to a destination IVR rule table 152 via connection 131. In addition, the routing engine 130 is connected to logic, at decision step 162, to determine whether a live agent is required, at decision step 162 for further processing of calls being handled. Where an agent is required, processing continues from decision block 162 to CTI 140, and where an agent is not required, processing continues to processing block 135 for an application to application connection to a designated destination IVR application 180.

[0009] Based on a caller profile associated with the telephone number and based on data retrieved from the CRM database 142, CTI 140 may set an optional flag for the call that

is provided in the agent queue. If a flag is set and the queue is long, the call may be bridged to a selected audio clip stored in the audio content library database 154. A customer with a particular call profile may then receive product information deemed to be interesting to the caller while the caller is waiting in the queue. When a call proceeds to the queue and is routed to an agent terminal 147, a screen pop may be displayed at the agent terminal 147 which is initiated by CTI 140. The screen pop may contain specific information associated with the caller's telephone number and may be automatically populated on the agent screen based on customer information retrieved from the CRM database 142. An example of such information is a customer size category such as small business, individual or large business and preferred call treatment.

[0010] The disclosed system provides a VoiceXML based software-driven switchboard powered by multiple rule engines that routes calls from one source to another. The switchboard offers four basic programmable paths as follows:

Path No.	Name	Origination		Destination
1	D2I	callers dial Directly into the switchboard	→	Another IVR app
2	I2I	callers came from another IVR app	→	Another IVR app
3	D2A	callers dial Directly into the switchboard	→	Agent
4	I2A	callers came from another IVR app	→	Agent

[0011] Switching logic inside the system is downloadable over the web from a centralized data source containing business logic and/or rules using standard HTTP interface. This data source 116 may be managed by a relational database management system or emerging technologies such as XML server (or XML database). The business logic database 116 may include rules based on the caller's telephone number, time of day, type of services interest to them in the past, their personal profile, and the customer segmentation (mid-class, upper-middle-class, high-rollers, etc.). The business logic database 116 may also contain a location driven rule subset such that the switchboard can use location information associated with the call to determine whether the caller is physically closer to a particular facility. The location information may be acquired using a WIFI or wireless DSL network. For example, if the caller (while visiting the new city) asks for a top-10 movie, the switchboard can pre-load a list of the theaters nearby that

show that particular movie. the subsequent dialog can provide relevant information to the caller. Because the system may be built as a web server, an enterprise can choose to deploy multiple systems that can present the same “image” to all callers by using the trigger-based refreshing mechanism supported by VoiceXML Root Engine 100. Such a trigger could be a scheduled event such as loading new switching logic every Sunday at 2 AM or pulling a new order from a given URL every 30 minutes.

**[0012]** VoiceXML Root Engine 100 is activated on from its *idol* position by an incoming call. For any call coming from a traditional time division multiplexing (TDM) based telephone network, the TDM network will deliver a DNIS associated with that call. Once activated, the VoiceXML Root Engine 100 first checks if the DNIS is defined in the Valid DNIS Rule Tables 114 for that call period. If not, the switchboard can be pre-programmed to either 1) not to answer the call or 2) play a pre-defined announcement and then terminate the call. If defined, the VoiceXML Root Engine 100 may launch a request via Link 112 to be executed on a J2EE Application Server 100 which in turn triggers a set of business rules and logic stored in Database 116 or encoded in a run-time rule engine residing on the application server.

**[0013]** Based on the business rules/logic matching to the DNIS and telephone number (TN) of the caller, if it is available and confirmed via ANI (automated number identification) or by the incoming IVR, a set of IVR dialogs in the form of VoiceXML pages is sent via Link 108 to IVR Dialog Engine Process 120. Process 120 will then engage a voice dialog with the caller using a set of pre-built statistical language models (SLM) designed specifically for a given DNIS profile. If the caller’s request is fully understood by Process 120, it sends a message (Task ID) via Link 122 to Routing Engine 130. Routine Engine decodes the message and determine if the call should be best handled by a live agent. If yes, it tells the switchboard to connect the call to CTI 140.

**[0014]** CTI 140 first checks with Customer Relationship Management (CRM) system 142 using TN via Link 141 associated with the caller. CRM 142 may decide based on the contact history for this customer that a personalized message, for example for consumer customers, or special message should be played to the caller if the Agent queue is

relatively long for that time of day. For example, a large number of calls sitting in the Agent Queue for Billing may just want to find out their monthly account balance around a due date window. In this case, CRM 142 can form a text message and put it in a personalized queue process 102 for those selected callers. Process 102 may use advanced text-to-speech (TTS) technology to convert the text message into an audio file. When the caller's turn comes, Process 102 may play that audio file to the caller and send the call to a group queue process 104 identified by Routing Engine 130. Process 104 maintains in real time a Task-ID to Skill-Set mapping table and uses this table to send the call to a specific queue served by a matching skill-qualified agent that is available at that time.

[0015] Process 102 may make an advanced reservation via process 104 in order to reserve a slot in the agent queue identified by process 104. For example, for an agent queue, such as a billing queue, with an estimated waiting time of about 5 minutes, process 102 can customize the personalized message to be about 4 minutes and 45 seconds. Thus, after the caller finishes listening to their personalized message, they will be placed in their reserved position in the billing queue for a much short wait time before served by a live agent.

[0016] Unlike a personalized queue, for a group queue, such as a generalized billing queue connected via link 105, each of the callers in that queue will hear the same message/music or product announcement while waiting to be connected to an agent. A Task ID obtained by dialog engine module process 120 can be used to dynamically trigger a different group message stored in Audio Content Library 154. For example, if the switchboard suddenly receives a few thousands calls within a very short period of time that are related to broadband Internet connection, a special group message may be invoked for all the calls sitting in the repair queue.

[0017] When a CTI 140 detects an idle agent, the CTI 140 sends the call context data via link 146 to the agent's desktop 147 by populating various screen fields. Then, CTI 140 may play a chained message from either the voice recordings of the caller or a number of concatenated TTS-generated audio messages associated with the Task ID. This is known as whisper transfer because the caller will not hear these audio messages while the agent

is listening to them. Whisper transfer is particular useful when the data link 146 is not initially available for the implementation of the switchboard in an enterprise call center operation.

**[0018]** If Routing Engine 130 finds a destination IVR application 180 (e.g. IVRa in the drawing) that matches to Task ID, it will check, via link 131 the destination IVR rule tables stored in database 152. Based on the structured rules and the interface template found for that destination IVR application, the switchboard activates a software connector 135 which composes various text and/or audio messages in a format readable by the destination IVR 180. If the destination IVR 180 (e.g. IVRa) is another VoiceXML-based IVR application, connector 135 can use VoiceXML standards based application-to-application connection methods, such as <submit> to pass the call and then release the call from the switchboard. If not, connector 135 can physically transfer the call to the destination IVR 180 (i.e. IVRa) and at the same time pass the call history via in-band or out-band methods.

**[0019]** Referring to FIG. 2, further details of a call center system is shown. The system includes IVR applications 206, SIP end point 202, a telephony communication instrument such as phone 204, and various additional IVR applications 208. The communication inputs are received at module 210 which is a switchboard routing module and processing continues for the call treatment to telephony media processor 211 and SIP listener 212. The telephony media processor 211 may retrieve DNIS rule tables from database 114. An output from telephony media processor 211 and SIP listener module 212 is routed to decision logic 221 where it is determined whether the call is defined in the DNIS rule table. When the call is defined in the DNIS table, processing continues by accessing business rule logic database 116. Where the call is not from a defined DNIS, then for the unknown DNIS, processing is routed to module 213 to construct a starting document for a default page. The default page for module 213 is then forwarded to application server 240. Where the call is from a defined DNIS and after the business rule logic and rules have been retrieved from database 116, processing continues to module 234 where a starting document is constructed for pre-defined CGU types based on the DNIS, time of

day, day of the week, holiday and telephone number. The resulting page is then forwarded as message 225 to the application server 240.

[0020] Also shown in FIG. 2 is an interconnection 113 between the VoiceXML root engine module 100 and the IVR dialog engine module 120. The IVR dialog engine module 120 is used to retrieve a document at the system URL as indicated at 241. In addition, an XML version document 215 is constructed and then passed back to the VoiceXML root engine 100 for additional call processing.

[0021] FIG. 2 illustrates detailed processing logic being executed after a call arrives at the switchboard but before the first prompt is played by the IVR dialog engine 120. For calls coming from a SIP (Session Initiation Protocol) endpoint 202, Root Engine 100 has a built-in firewall to automatically accept or reject a corresponding http request based on their originating host IP address. If the access is granted, an http request (will wake up SIP listener process 212 which will then take a proper action. If calls come from a TDM telephone network, it will trigger one of telephony media processors 211 on the switchboard. Process 211 can be pre-programmed to reject certain unknown DNIS, such as by not answering the call at all. In addition to DNIS, the TDM network may also deliver ANI which may or may not correspond to the TN for the calling customer. For SIP calls, the TN is normally delivered as an argument attached to an http request, such as follows:

`http://IVRswitchboard.sbc.com/cctp_ACR_homepage.vxml?TN=5125551234&Type="Billing"&...`

[0022] If DNIS is not defined in Database 114, decision step 221 will activate process 213, which will construct a VoiceXML document containing the default dialog: CGU(<default>). This default dialog tries to reach customer goal understanding (CGU) without knowing where the call came from (since an associated DNIS is not defined in database 114.)

[0023] If DNIS is defined link 216 or SIP call link 214 is determined to have originated from an authorized IP host, the root engine 100 will immediately fetch a starting



document 215 from a known web server, or use a cached version. A first actionable tag <vxml> inside document 215 may trigger the switchboard system to fetch a root document written in VoiceXML and then pass it to a built-in VoiceXML run-time interpreter which then executes the processing logic specified in this root VoiceXML document.

[0024] If the DNIS is not on the excluded list, process 211 will search Database 114 to determine if there are any special rules associated with this DNIS. Decision 221 may consult Database 116 to determine which CGU dialog should be loaded based on the DNIS/URL and TN. Once a match is found from Database 116, this root document may instruct the switchboard to activate a software constructor 234. Constructor 234 may run on a standard J2EE Application Server. The input link 224 to constructor 234 contains business rules and processing logic defined for a group of DNIS/URL (e.g., all the DNIS associated with a single billing number for that enterprise's customers) and TN if available (from either the caller's ANI or passed from an incoming IVR application (e.g., IVR2 as shown in the drawing)). Based on a set of pre-defined CGU dialog templates, constructor 234 builds in real-time a set of dynamic VoiceXML pages and store the first page at a pre-defined location according to document 215. After that, Root Engine 100 returns control to the switchboard. After the return, the VoiceXML interpreter continues to process the first VoiceXML form (id="CGU") specified in the starting document. The tag <submit> inside the VoiceXML form CGU will cause the switchboard to activate IVR Dialog Engine 120 by fetching the page just generated.

[0025] FIG. 3 illustrates a method of routing a call and providing follow-up electronic notifications. As shown, a call is received at an automated call system, at 302. An example is a call received at an incoming IVR. A call evaluation based on business logic is then performed, at 304. Based on the call evaluation, the call is routed to a destination IVR or to an agent queue, at 306. The call is then handled by the agent or the destination IVR. In certain situations, a follow-up notification to the caller is desirable. An example is a customized marketing promotion that is matched to certain characteristics of the caller or the subject matter discussed by the caller during the call to the IVR. Based on the call and the caller profile, an electronic notification may be scheduled for delivery at a

selected time to send an email to the caller. The email may include a targeted communication message, such as an advertisement or other promotion, that relates to the subject matter of the call processed by the IVR at the call center system, as shown at 308. With this method, permission based targeted marketing programs using email may be automatically distributed and scheduled for callers to a call center. For example, the caller may have a problem that is not solved by a live agent or the destination IVR, such as a customer request for certain information. The electronic notification system may be used to send an email to the caller with the information requested by the caller. The disclosed method may allow for follow-up caller service, enhanced customer care, and cross selling capabilities.

[0026] Referring to FIG. 4, further details of methods of using an IVR dialog engine module 120 and the call center system are shown. Inputs including the model VoIP telephone number and other data retrieved from VoiceXML routing engine module 100 is received as inputs 113 at the module 410. At 410, specific CGU dialog modules are loaded and session records are stored at the session record database 426. A CGU record is processed at 420 and compared to decision threshold logic, at 421. An initial threshold may be a 20% decision threshold and may require retrieval and access to offline tuning data 424. A CGU confidence determination is made at decision step 422. If the CGU confidence level is low, then error handling module 430 is accessed which may result in a bail out of the process at 460. If the confidence factor at 422 is a medium level, then the CGU is loaded and direct dialog models are handled at 414 and 440. A DDM confidence factor is then checked at decision step 444. Where the CGU confidence level is high, a confirmation occurs at 450 and the routing engine is instructed with message 402 to handle routing of the call, via routing engine 130. Referring again to decision step 444, if the DDM confidence level is low, a bail out occurs at 448 and the call transfer module 326 is accessed to complete the call. If the DDM confidence level is a medium level and a CGU level is loaded at 442, the processing continues at 422. In addition, the call transfer module 468 provides instructions 469 to the CTI 104 as described above.

[0027] As shown in FIG. 4, process 410 reads a call profile created dynamically by VoiceXML Root Engine 100 for each call. The call profile specifies which CGU dialog modules should be used for this caller (if TN is available) or for any caller dialing into a group of DNIS numbers associated with a published number representing a business category such as consumer-billing or consumer-repair. In addition, the profile may also contain control codes from an incoming IVR so that the CGU dialog modules can continue a dialog with the caller in an efficient and user-friendly manner. For example, if prior to being sent to the switchboard, the incoming IVR application (e.g., IVR<sub>1</sub>) already asked the caller to select a language (English, Spanish, or etc.), the IVR application will have an option to create a special control code (e.g., LANG=1 where '1' indicates English as a preferred language by the caller) and then pass the control code with other session variables, such as the TN via a shared session records database 426.

Alternatively, an incoming IVR application can choose to send the control codes via an in-band signaling method (such as channel associated signaling (CAS)) to trigger a dual tone multi-frequency (DTMF) reader inside process 410. The DTMF reader reads the control codes generated from the incoming IVR application and uses these codes to modify the logic of the CGU dialog modules accordingly. For example, if the caller already selected the language while being connected to the IVR application, the CGU dialog will not ask the same language question again.

[0028] Process 420 employs a selected CGU dialog (most often a highly focused CGU dialog designed specifically for a certain profile such as consumer-billing or business-payment etc) to gather further information about the caller's goal (why they are calling and so on). If the CGU dialog is able to recognize their goal, process 420 sends a confidence score of that recognition to process 421 which controls the decision threshold. The decision threshold may be periodically changed based on an offline tuning database 424. For example, when the switchboard is initially installed, database 424 may only contain the tuning data collected from a few thousands of calls. At that initial stage, process 421 may choose to use a very tight threshold to determine if the confidence score from a CGU dialog is high enough to be acceptable or not. As the size of Database 424 grows over time, the decision threshold may be automatically adjusted to achieve the

balance desired (between misunderstanding a customer's goal and rejecting the assessed CGU).

[0029] Based on fuzzy logic inside process 420, decision 422 will make a final determination as to whether the confidence score from the CGU dialog is low, medium, or high. If the confidence score is high, IVR Dialog Engine will invoke confirmation module 450 which will inform the caller 1) the goal recognized in terms of a set of pre-defined user tasks, and 2) what the caller can expect next — to be sent to another IVR application or to an agent queue. Then, IVR Dialog Engine will send the task ID along with other session data via link 402 to routing engine 130.

[0030] If the threshold is medium, process 414 will load a set of specific direct dialog modules (DDM) based on the outcome of the CGU Dialog Process 420. Such an outcome is represented by a group ID so that DDMs (<ID>) loaded have a focused conversation with the caller to further clarify what they really meant from what they said. For example, if a CGU (<Consumer:Billing>) is not determinative of whether the caller really wants to a) get information about their bill or b) give information about their bill, block 414 can load the two DDMs and then pass the control to process 440. If DDM process 440 is able to successfully recognize the customer's goal with a high confidence, the caller will be sent to confirmation process 450. If decision 444 receives a medium confidence score from process 440, it may choose to load a different CGU dialog and start the customer goal determination process again and by pass the control to Process 420. For example, the DDMs used in Process 440 may reach a conclusion that the call is not really about billing. Instead, what the caller really wants is to pay their bill. If that is the case, a CGU (<Payment>) will be loaded.

[0031] If decision 422 considers the confidence score from the CGU dialog process 420 to be low (or a failed recognition generates no confidence score at all), a set of error handling modules (EHM) will be invoked to recover from such a situation such as giving the caller another chance or simply bailing out to process 460, via the call transfer module (CTM).

[0032] Depending upon the useful information collected during such a call interactive problem determination session, CTM 468 may decide to set various flags for playing a personalized message when the call is sent to a personalized queue. In addition, CTM 468 can be configured to set a flag for various agent queues such as billing, repair, and etc. When such flags are set, the caller may hear a customized message common to all the callers in that agent queue (instead of meaningless music-on-hold). If the last recognition event (from CGU dialog or DDM dialog) produced an audio recording of the caller's utterance, CTM 468 may compose an appropriate audio message for the agent to listen to. Such an audio message may start with an introduction tone followed by the caller's utterance(s) selected and followed by an ending tone. If there is a confirmed recognition result from the session, a corresponding text message will be formatted to feed CTI 140 via link 469 for a proper screen pop at the agent's desktop. Finally, CTM 468 has an option to prepare a text message related to the session so that the message may be used for an outbound E-channel delivery such as an email notification.

[0033] Referring to FIG. 5, further details of the routing engine module 130 are shown. The routing engine module 130 is coupled to application server 170. Additional inputs received from the IVR dialog engine 120 are received as a task ID, telephone DNIS, and session data 122 at the business rule and logic processing module 531. The task ID, telephone DNIS, and telephone number are passed to each notification module 515 and may be used to access the business rule logic database 116. At decision block 502, the task ID is compared to determine whether agent handling is appropriate. When agent handling is not needed, a particular IVR application is identified at 520 and destination IVR rule tables 152 are accessed. When IVR identification is retrieved, then processing continues to the decision step 525 to determine if the IVR application is VoiceXML compliant. If so, processing continues to set-up the application to application transferring module 530. In the case where the task ID is best handled by an agent, processing continues at module 526, which accesses a call transfer module. A call transfer module 526 performs steps such as setting a flyer for personalized messages, sending a flag for a category queue, preparing audio messages for the agent, preparing text messages for a screenpop via a CTI, and preparing data for outbound e-channel delivery. Processing is

then followed by instruction 521 being communicated to CTI 504 to further interface with the agent terminal.

[0034] In the situation where the IVR application is not VoiceXML compliant, at 525, then a determination is made as to whether the IVR application supports an external data interface. If not, processing continues at module 550 to search a dial-through table for the IVR. Thereafter, a three-way call transfer may be made at 570 and routing to a final destination IVR 180 is handled. If the IVR application does support an external data interface, then delivery of session data to the application occurs, at 540, and a two-way call transfer is handled, at module 560, prior to final routing to the destination IVR 180.

[0035] In the situation where the IVR application is VoiceXML compliant, at 525, once a pre-defined task is matched during the dialog with the caller (and confirmed), IVR Dialog Engine sends the applicable session data such as <Task-ID>, TN, and DNIS to process 531. Process 531 searches Database 116 to determine how the call should be handled given the business logic or processes associated with such a task. The search process link 109 can be executed by a rule engine which typically resides on a J2EE application server. As a part of the execution logic, decision 502 will determine whether the task should be handled by a live agent instead of being routed to another IVR application. If Yes, the control is passed to process 526 (CTM).

[0036] If not, process 520 is invoked to identify a proper IVR (destination IVR-ID) application to be routed and then uses <IVR-ID> to search a database containing destination IVR profiles (Database 152) via link 510. Based on the profile pre-defined for the destination IVR (IVR-ID), decision 525 determines if the IVR is VoiceXML capable. If yes, the caller is essentially sent to another VoiceXML application through process 530 using standards-based application-to-application transfer methods such as <submit> which can easily pass all session related data to the VoiceXML-based IVR application via Link 531.

[0037] If the destination IVR is not VoiceXML capable, decision 535 determines whether the IVR is capable of supporting any external data interface to receive the session data. If yes, process 540 is invoked to deliver the session data to the destination

IVR via a shared database or using a specific software adapter built specifically for that IVR. After delivering the session data, process 560 is invoked to ask the switchboard to make a two-way transfer by making an outbound call to the destination IVR first and then bridge the call that has been put on hold after confirmation process 450 finally disconnects itself (the switchboard) from the call.

[0038] If the destination IVR does not support any external data interface, Process 550 will search if there is any “dial-through” table defined for the IVR from Database 152. If found, Process 550 constructs a single audio file (.wav file) to contain a series of DTMF commands based on the session data collected. For example, if the destination’s main menu asks the caller to enter a telephone number (TN) after DTMF-1 is received and then a sub-menu, the audio file generated by process 550 for such a short cut may have a following structure:

Audio Segment	Silence of 3 seconds	DTMF-1	Silence of 1 second	DTMF: <10-digit TN>
Purpose	<i>Non-interruptible prompt</i>	Menu Selection	Safety buffer	Enter TN

[0039] After such an audio file is constructed, Process 570 is invoked to make a 3-way call transfer. First, the switchboard puts the current call on hold and then makes an outbound call to the destination IVR. Upon the connection, the switchboard plays the audio message containing the short-cut sequence which will guide the destination IVR to a proper menu before connecting the original caller. This method is also known as whisper transfer since the caller will not hear the audio message played. At the end of the playback of the audio message, the switchboard bridges the call to the 3-way connection and then disconnects itself immediately.

[0040] The disclosed system uses Voice Extensible Markup Language (VoiceXML) based technology in general and focuses particularly on improving the capability of switching traffic in and out of a centralized call center environment based on changing business needs and dynamics of existing and future IVR applications. The software architecture is particularly applicable to the call center operations of large enterprises where many of their existing IVR applications are written in proprietary programming

languages and have to co-exist with VoiceXML-based IVR applications. Rule engines used to power the system may be assembled from various light-weight modules that are specially designed for real-time responses when callers are switched in and out of the switchboard from one IVR to another.

[0041] The above disclosed subject matter is to be considered illustrative, and not restrictive, and the appended claims are intended to cover all such modifications, enhancements, and other embodiments which fall within the true spirit and scope of the present invention. Thus, to the maximum extent allowed by law, the scope of the present invention is to be determined by the broadest permissible interpretation of the following claims and their equivalents, and shall not be restricted or limited by the foregoing detailed description.